

App. No. 09/682,065

Response to objections

1. Number (4.) of the Second Office action states that the following quotation from 35 U.S.C. § 103(a) forms the basis for all obviousness rejections set forth in said second Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious *at the time the invention was made* to a person having ordinary skill in the art to which said subject matter pertains.

Number (5.) furthermore states that

Claims 23-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Natori et al. (U.S. 2004/0060036 A1) in view of Bohrer et al. (U.S. 6,106,569).

The main reference on which the 2nd Office Action is based is U.S. Patent Application number 2004/0060036 A1, to Natori et al, of Kabushiki Kaisha Toshiba, Japan.

Referenced U.S. Patent Application 2004/0060036 A1, to Natori et al, was originally filed at the United States Patent and Trademark Office on September 15, 2003 (number 10/881,474), while Applicant's application 2003/0014560 A1 (number 099/682,065) —which is the subject of this response— was filed on July 16, 2001. Thus, Applicant respectively argues it is not at all possible that such alleged obviousness, as exposed in the quotation above, could have taken place as the Natori et al Patent Application does not predate Application's date of invention.

Nonetheless, and independently from the previous argument, we notice that in the spirit of the opening rejections (number 6.) there appears to be an inclination to identify a certain correspondence between several terms used in the texts of our patent application, more specifically in the Claims section, and the expressions used by Natori et al., such as is exemplified in the following table:

App. No. 09/682,065

| 2003/0014560 A1 (Our application) | 2004/0060036 A1 (Natori et al) |
|---|--|
| CORE dimension | an enterprises system framework |
| a plurality of applications | a client program |
| relating the core dimension to the application | constructing an enterprise system using the enterprise system framework |
| the CORE is a complex organizational referee engine which manages information and interaction between different extensions of the architecture attached to it | a client/server application system framework, a web application system framework, a server application system framework |
| is a complex compound of software components that exchanges information with other instances of itself, within the core dimension related | the delivery of data between systems |
| the CORES within the core dimension share information and integrate the system architecture | The enterprise system basic framework ... communication processing components |
| the applications | a client/server application system group, a Web application system group and a server application system group |
| abstractions | a client/server application system framework 12, a web application system framework 13, a server application system framework |
| drivers, abstraction layers and a unique CORE | abstract class, a subclass, and customized sub class by materializing the abstract method of the subclass and adding a new attribute or method |
| a plurality of terminal devices | Clients, Web application server, Internet server, Database Server |
| the abstraction layers consist of a software layer that hides implementation details and data structures of a specific software | The client/server application ... "abstractly defines basic attributes and behaviors of a stand-alone client/server application system, and is expressed as aggregate of abstract classes and concrete classes"; sections 0061-0063. |
| the module extensions are comprised of an abstraction layer and a driver | abstract class, customized subclass |

The Applicants argue that, even if these expressions may apparently convey similar meanings, in the majority of cases they actually mean different concepts, which even may be vastly different, and hence are applicable in their respective contexts only, so both sides in this collection of pairs of expressions may not be readily regarded as being equivalent. The subjectivity of the interpretation of these expressions is probably augmented by the high level of abstraction which is noticeable in the texts of all disclosures and claims for each of the inventions involved.

App. No. 09/682,065

Beginning with No. (7.) the Office Action exposes similarities between our patent application and Bohrer et al (U.S. Pat. No. 6,106,569), and in a similar fashion finds corresponding terminology used in both writings. In the USPTO database these are some 3,476 issued patents related to the term "Object-oriented programming." The Applicants argue that undoubtedly most of these inventions must by force employ a common language, even striving to standardize the use of terms, concepts and definitions in a common, mutually accepted terminology (or specialized jargon), that is what standards are about. Many inventions regarding techniques, tools, resources, and products such as object-oriented Database systems, languages, application builders, design tools, prototyping and modeling tools, debuggers, application systems testing methodologies and tools, et cetera, all depend on the use of a common, consistent language in order to convey the correct meaning to the community of professionals for which they are intended. The concepts involved are the common ground of Object-Oriented Programming, or OOP, which has caused a *paradigm shift* in applications development, as it is regarded to be more flexible, easing program development and maintenance, and consequently is widely popular in large scale software engineering. A characteristic of object-oriented programming methodology is that the operations of any interface of a class are usually independent of each other, meaning that an interface places no requirements for clients to invoke the operations of one interface in any particular order. The benefit is that client program code can rely on the fact that the operations of an interface are available for use whenever the client holds a valid reference to the object. Another beneficial consequence is that a client-server (or layered) design results, where servers do not depend in any way of the clients.

Beyond any doubt, the previous art inventions mentioned by the examiner, and the present invention too, belong in the field of Object-Oriented programming. The fact that similar terms, expressions, and concepts appear in their respective texts, in our opinion does not constitute enough reason to invalidate the novelty of said respective inventions. On the contrary: in all cases many a significant contribution may be identified toward making such concepts and techniques evolve into useful tools and solutions, an endeavor which is the marrow of the motivations for creating an invention.

In a fashion similar to Number (6.), arguments included in Number (7.) include the following apparent equivalences:

2003/0014560 A1 (Our application)

U.S. Pat. No. 6,106,569 (Bohrer et al)

| | |
|---|---|
| Application dimension | Software applications |
| CORE dimension | Base layer, Common Business object layer, Common Business process layer |
| relating the application dimension to different applications | Applications, software applications |
| The CORE that can accept different kinds of software extensions dynamically in a hot plug fashion | The request support provided by the concrete extensions is cumulative, i.e. they provide a dynamic mix-in capability. |

However, in our invention, the definition of the CORE (*Complex Organizational Referee Engine*) is conceived so as to perform specific functions that are different from Bohrer's defined terms of "Base layer", "Common Business object layer", and "Common Business process layer". Similarly, so is the "hot plug" fashion capability in our invention whereby the CORE can accept and integrate software extensions dynamically.

In number (11.) it is stated that *"Bohrer teaches the Core Business Process layer 103 is an abstract layer (layer 103 does not provide executable code; col. 6, lines 62-67), and contains the basic functions which*

App. No. 09/682,065

all of the application programs, and each one is built for one specific type of application (col. 6, line 67 - col. 7, line 14), and interfaces with a CORE"

Bohrer's disclosure actually says: *"The layer 103 above the Common Business Objects layer 102 already comprises Core Business Processes and can be regarded as the Core Business Process layer 103. Although layer 103 usually does not provide executable code, within this layer 103 the business software applications developed using the present invention begin to take shape."*, and in the next paragraph it says *"The resulting framework is constructed in a way to contain commonly used functions as well as to be easy to extend."* Again, Bohrer's concept of "Core Business Processes" must not be confused with, nor equaled to, our concept of *CORE - Complex Organizational Referee Engine*, as these are not equivalent terms: Bohrer specifically explains: [quote] *"Each Core Business Process layer is built for one specific type of application, as for example General Ledger or Warehouse Management"* [end quote] — whereas our definition of CORE, which is a control logic software kernel, a key resource allowing for the construction of a conceptual model composed of various drivers, abstraction layers and a unique core that provides support by playing a central role between different extensions of an application, and enabling inter-application communication.

In number (16.) it is stated that *"Brobst teaches replace a terminal device with a new terminal device consisting of the adding the step of changing the driver for the extension (CF extensions need change to accommodate new operating systems; col. 5, lines 12-25)."*

In number (17.) it is further stated that *"Brobst teaches add a new terminal device to a system using the method in claim 1 consisting of adding the steps of*

a) constructing a new extension for the terminal device (CF extensions need to change; col. 5, lines 12-25),

What Brobst in fact says is [start quote]:

"In this way, the server program can interface with multiple operating systems because only the CF extensions need change to accommodate new operating systems, while the server program can remain the same."

[end quote]

...where 'CF' is defined as [start quote]:

"the core framework (CF) is a set of base business application classes that are shared by multiple development programs."

[end quote]

We believe that the concept inherent in our definition of "terminal device" (Detailed Description 36: "A terminal device refers to any kind of software, hardware device or terminal user that performs a specific activity. For example: a DBMS storing data, a Control network, etc."), though using a similar wording, in fact differs from Brobst's CF extensions. The need to replace an extension does not arise solely in order to accommodate a new operating system or to create a platform independent system, though such a need is indeed satisfied successfully by the proposed architecture of the invention.

A similar reasoning may be applied to (b) and (c) in number 17:

b) interfacing the new extension into the CORE (add desired extensions to the server framework itself; col. 5, lines 27-29);

c) asking the CORE for the required data and information to handle the new extension (When such changes are determined ... at box 106, col. 5, lines 52-58)."

Adding new extensions to an application is a common routine performed upon object-oriented systems, and as such is mentioned as a feature of both Inventions.

App. No. 09/682,065

REMARKS - General

By the above amendment, Claim number 23 has been modified into Claim 31 based on the Section 112 rejections and to stress its particular terminology, in order emphasize the novelty of the invention from the outset, and to define the invention more particularly and distinctly so as to overcome technical rejections and define the invention patentability over the prior art.

The new claim and the reasoning in our response above promote a novel and unique invention that is not obvious to someone skilled in the art even in light of Natori, et al. US2004/0060036 in view of Bohrer US 6,106,569, and even in view of Brobst et al (U.S. 5,893,106)..

Conclusion

For all of the above reasons, applicant submits that the specification and claims are now in proper form, and that the claims all define patentability over prior art. Therefore the applicant submits that this application is now in condition for allowance, which action is respectfully solicited.

Respectfully submitted,



Jeffrey M. Furr, Esq.
Registration No. 38,146

I hereby certify I have transmitted this paper by fax to the Patent and Trademark Office at 571-273-8300 on September 12, 2005.

September 12, 2005.



Jeffrey M. Furr, Esq, Reg. No. 38,146.